

Prácticas de Robótica

Práctica 3 - *Tracking*.

Seguimiento basado en visión

Objetivos

1. Implementar funcionalidad para **seguimiento de un objeto** basado en visión
2. Mantener la odometría en funcionamiento durante el seguimiento
3. Coger el objeto cuando este en la posición adecuada para el robot

En moodle encontrareis los ficheros base para esta práctica. **Para evaluar las entregas de las prácticas los profesores asumen que el “signature” de las funciones requeridas es como en dichos ficheros. Si se cambian por algún motivo, comentar con los profesores y añadir un README explicando dichos cambios.**

Descripción

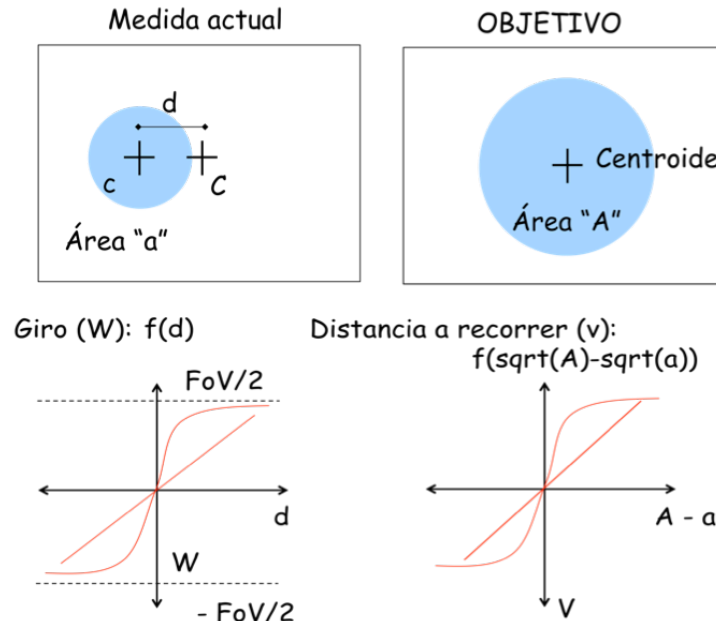
1. Calibrar posición de la cámara, colores, y tamaños/posiciones del *target*

Tomar unas cuantas fotos de la **pelota roja** disponible a distintas distancias y posiciones, lanzar con ellas el script `get_color_blobs.py`. Analizar qué rangos de color, área y posición os parecen más adecuados para que sean vuestro “*target*” mirando el visor que se abre al lanzar dicho script.

2. Implementar funcionalidades para **seguimiento y captura de pelota roja**:

Implementar una funcionalidad nueva en la clase Robot, para seguir un objeto, que puede estar parado, o en movimiento. Analizar la relación entre la diferencia de áreas y el “offset” respecto al centro de la imagen, y las velocidades lineal y/o angular que necesitamos.

```
# ADD to the Robot class a method to track an object, given certain
parameters
robot.trackObject(colorRangeMin=[0,0,0],    colorRangeMax=[255,255,255],
                  targetSize=??, target??=??, ...)
```

**Requisitos de esta funcionalidad:**

- El robot debe mantener la odometría actualizada constantemente. Esta tarea se lanzará como en la práctica anterior (ver **p3_base.py**).
- Para que sea correcto, al realizar el seguimiento, el robot no va a trompicones, no se tropieza con el objeto, ...
- El robot para cuando está a en la posición adecuada para coger la pelota, y la coge.

Documentación y Links de interés

Para realizar esta práctica se recomienda el uso de la biblioteca openCV (como se indica en los ejemplos de `get_color_blobs.py` y `p3_base.py`) y, al menos para la primera versión, la función "SimpleBlobDetector" de OpenCv:

https://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_feature_detectors.html?highlight=blob#simpleblobdetector

<https://github.com/opencv/opencv/blob/master/modules/features2d/src/blobdetector.cpp>

Este ejemplo describe en bastante detalle las opciones de esta función <https://www.learnopencv.com/blob-detection-using-opencv-python-c/>

Trabajo PREVIO (se evaluará al entrar a la práctica en un cuestionario)

- Que funciones/estrategias vais a aplicar para decidir la nueva v y w en función de los parámetros medidos en cada nueva imagen/frame? Cuales serán dichos parámetros?

- Preparar **pseudocódigo/esquema en python** para estos dos métodos de la clase **robot**:

```
def trackObject(self, colorRangeMin=[0,0,0], colorRangeMax=[255,255,255]):
    targetSize=??, target??=??, catch=??, ...)
    finished = False
    targetFound = False
    targetPositionReached = False

    while not finished:
        # 1. search the most promising blob ..

        while not targetPositionReached:
            # 2. decide v and w for the robot to get closer to target position

            if ...
                targetPositionReached = True
                finished = True

    return finished

def catch(self):
    # decide the strategy to catch the ball once you have reached the target
    position
```

Evaluación

- **Entrega del código, en moodle, el día establecido para la entrega de la tarea.**

Se valorará:

- Claridad y legibilidad del código. Al menos se recomienda incluir un “[docstring](#)” en cada una de las funciones principales.
- Funcionalidad correcta de las funciones requeridas
- **Demo** a los profesores de las tareas realizadas (todo el grupo tiene que estar presente en la demo, y se realizarán preguntas a todos los componentes sobre la entrega). Durante la sesión si se terminan, o al empezar la sesión de la práctica siguiente.